This document is part of the Project “Machine Translation Enhanced Computer Assisted Translation (MateCat)”, funded by the 7th Framework Programme of the European Commission through grant agreement no.: 287688.

Machine Translation Enhanced
Computer Assisted Translation

D5.1 - Baseline Definition

Author(s): Alessandro Cattelan, Marcello Federico.

Dissemination Level: Public

Date: April 27th, 2012

Grant agreement no.	287688
Project acronym	MateCat
Project full title	Machine Translation Enhanced Computer Assisted Translation
Funding scheme	Collaborative project
Coordinator	Marcello Federico (FBK)
Start date, duration	November 1st 2011, 36 months
Dissemination level	Public
Contractual date of delivery	February 29th, 2012
Actual date of delivery	April 27th,2012
Deliverable number	5.1
Deliverable title	Baseline Definition
Type	Report
Status and version	Final V1.2
Number of pages	37
Contributing partners	Translated, FBK
WP leader	FBK
Task leader	FBK
Authors	Alessandro Cattelan, Marcello Federico
Reviewers	Christian Buck
EC project officer	Alexandra Wesolowska
The partners in MateCat are:	Fondazione Bruno Kessler (FBK), Italy
	Université Le Mans (LE MANS), France
	The University of Edinburgh (UEDIN)
	Translated S.r.l. (TRANSLATED)

For copies of reports, updates on project activities and other MateCat-related information, contact:

FBK
MateCat
Marcello Federico — federico@fbk.eu
Povo - Via Sommarive 18 — Phone: +39 0461 314 552
I-38123 Trento, Italy — Fax: +39 0461 314 591

Copies of reports and other material can also be accessed via http://www.matecat.com

© 2012, Alessandro Cattelan and Marcello Federico
No part of this document may be reproduced or transmitted in any form, or by any means, electronic or mechanical, including photocopy, recording, or any information storage and retrieval system, without permission from the copyright owner.

Executive Summary

This report is organized in two main sections and two appendixes. The first section describes the standard CAT workflow adopted by the translation industry, introduces the most popular commercial CAT tool (SDL Trados Studio) that is taken as reference in our project and, finally, it outlines user and functional requirements of the MateCat Tool. The second section reports on the first field test carried out to measure translation productivity with SDL Trados Studio powered with a commercial translation memory (MyMemory) and a commercial machine translation engine (Google Translate). Finally, the two appendixes provide, respectively, revised specifications of the software architecture of the MateCat Tool and a road map for the development of its first version.

Table of Contents

Executive Summary **3**

1 MateCat Tool Requirements **6**

1.1 Standard CAT Workflow in a Nutshell **6**

1.2 Working with SDL Trados Studio **9**

1.3 Working with the MateCat Tool **10**

1.3.1 Self-tuning MT 11

1.3.2 User-adaptive MT 12

1.4 Conclusion **12**

2 First Field Test **13**

2.1 Scope and limitations **13**

2.2 Defining the KPIs **14**

2.3 Field test data **14**

2.4 Data collection setup **15**

2.5 Data collection issues **16**

2.6 Filtering procedure **17**

2.7 Results on KPI 1: Post-editing Effort **18**

2.8 Results on KPI 2: Time to Edit **20**

2.9 Discussion **21**

2.10 Conclusion **22**

3 Appendix 1. Architecture Specifications **23**

3.1 MateCat Tool Architecture **23**

3.2 Context Manager Module **25**

3.3 MT Module **26**

3.4 Use Cases **27**

3.4.1 Project Manager 27

3.4.2 Translator 28

4 Appendix 2. Development Road Map **28**

4.1 MateCat Tool v.1 ..... **28**

4.1.1 Features 28

4.1.2 Schedule 29

5 Appendix 3. Data collection and post-editing guidelines **30**

5.1 Issues in data collection 30
5.2 post-editing guidelines 30
5.2.1 General information 30
5.2.2 Note for the Legal translation jobs 31
5.3 Step by step guide 31
5.3.1 Installing the plugin 33
5.3.2 Key generation 33
5.3.3 Project creation in Trados Studio 34

1 MateCat Tool Requirements

The MateCat Tool will operate in the well-established Computer Assisted Translation (CAT) workflow. The main goal of this report is to outline the assumed operating conditions, functional requirements and architecture design of the MateCat Tool. In order to achieve this goal, we first overview the standard CAT work flow based on currently available technology. In this way, besides introducing basic concepts of the CAT framework, we will also outline the features of a commercial CAT tool, SDL Trados Studio, which will be used as reference for the MateCat Tool.

1.1 Standard CAT Workflow in a Nutshell

Language service providers (LSP) manage most of their work through projects, i.e. the translation, review, proofreading and editing of one or more files carried out at the client's request by an agreed deadline.

The first phase of any translation project is the volume analysis, which helps to determine the resources needed to complete the project. The basic unit most commonly used to gauge the text to be translated is the "word", be it "source word", "target word" or "equivalent word":

- **Source words**: total number of words contained in the original files, regardless of repetitions or matches from the TM;
- **Target words**: total number of words contained in the translated files, most commonly used for files where the source words cannot be counted (i.e. scanned PDF files);
- **Equivalent words**: number of words calculated by assigning different percentage values to new words, repetitions, identical matches or fuzzy matches (a similar, but not identical, match to the source segment found in the translation memory). Typically the "equivalent word count" is used when analyzing files using a CAT tool or other dedicated tools that can leverage previous translations and repetitions. A "new word" is counted as one word, whereas repetitions and identical matches are typically counted as 30% of a word and fuzzy matches are assigned a specific value depending on their quality (i.e. how close they match the translation found in the TM). The value of the repetitions, identical matches and fuzzy matches is usually determined by the LSP.

The volume analysis allows LSPs to estimate the resources needed to complete the project. When translating with CAT tools, volume analysis is usually based on the equivalent word count so as to leverage available translation memories provided by the end customer or maintained by the LSP. The common assumption is that the standard translation output of a professional translator is around 2,000 equivalent words per day. Based on this assumption, LSPs estimate the number of translators needed to complete the project by the deadline agreed with the end customer.

Machine translation output is rarely considered in volume analysis, as there isn't an industry-accepted method for measuring MT productivity gains. Being able to effectively measure such productivity gains would make it possible to prepare more accurate volume analyses which account for potential reduction of the required work in terms of professionals to be involved, time to complete the project and total cost.

Project managers assign a set of files and resource materials (translation memories (TM), glossaries, style guides) to the translator or team of translators working on the project. Exchange of source files, resource materials and other relevant information is most commonly carried out via email or dedicated FTP servers. With server based translation memory systems, translators can connect to a centralized TM which is set up and maintained by the end customer or the LSP and updated in real time with translated segments from all translators working on the project. Resource materials are used to guarantee that the appropriate terminology and style is used throughout the project and consistency is maintained. Also, common resources such as TM and glossary or a terminological database (term base) allow the translators to work more quickly thus reducing turnaround times.

This process, however, often implies that translators constantly consult several resources at once: local (and server-based) translation memories, glossaries maintained in CSV or Excel format, term bases (web-based, file-based or integrated in the CAT tool), style guides provided as separate documents. When working in teams, translators may also be required to exchange relevant information with other translators or project managers, usually via email. The many resources that need to be consulted and the need to manually exchange information can prove an issue in complex projects leading to loss of information (translated segments or terminology) and less time available for the translation.

Most CAT tools break the source text into smaller units called segments, i.e. chunks of text corresponding to paragraphs or sentences and usually delimited by punctuation marks. Before presenting the segment to the translator, the CAT tool matches the content of each seg-

ment to the source segments contained in the TM. When an identical match (100% match) is found, it is automatically displayed in the translation editor - translators then check the quality of the match and either confirm the translation without further editing or correct it as they see fit. Moreover, CAT tools usually provide fuzzy matches, i.e. translations segments from the TM that partially match the source segment.

Fuzzy matches are ranked using tool-specific algorithms which assign a percentage score to each match based on its similarities to the source segment such as identical sequence of words in the source and target segments, same words with different ordering, morphological variants of the same words (e.g. conjugated verbs, plurals, etc.), spelling variants or errors, etc. The fuzzy match percentage can then be altered using a fixed penalty system which assigns a penalty score based on certain patterns, for instance penalties can be applied when the matching segment contains formatting, tags, numbers.

Machine translation matches can either come from the translation memory database (where they are identified using specific markers - e.g., the "MT!" attribute) or from an MT engine connected to the CAT tool. Matches from the TM and from the MT are ranked together using the same fuzzy matching algorithm. Since machine translated segments match exactly the source text while generally being of lower quality as compared to a 100% match from the TM, a penalty system is used to account for the quality difference. The MT penalty score is usually 15% and MT matches are assigned a fuzzy match percentage score of up to 85% (matching percentage can be lower due to penalty system).

Moreover, translators can perform searches on the translation memory database using the concordance feature. This feature allows translators to search for a sequence of words (fragments of text, sub segments or multiword terms) in the translation memory database in order to obtain suggestions for terminological or stylistic issues. Concordance results can match partially or exactly the search pattern and are ranked similarly to fuzzy matches.

Translators translate one segment at a time and make use of the automated suggestions provided by the CAT tool in the form of matches. Each time they complete the translation of a segment this is saved in the TM for reuse. When all segments have been translated, the translator or a reviewer can further edit the translation and the TM is updated with any changed segment.

1.2 Working with SDL Trados Studio

The CAT tool used as a reference for the MateCat Tool is SDL Trados Studio, a standalone CAT tool derived from the traditional Trados suite, one of the first and most popular tools on the market that is based on the translation memory technology. It must be installed locally and only works under the Windows operating system.

SDL Trados Studio is designed to be used by one user at a time. However, it offers some features such as the possibility to connect to a centralized TM server and to exchange project packages containing information about the status of a segment (translated, reviewed, approved etc.), which make it possible for different users to collaborate (translator, reviewer, project manager).

Contrary to previous versions of Trados, in which individual files are translated in either a standalone editor (Tag Editor) or in Microsoft Word, in SDL Trados Studio "all files are translated and managed as part of a project. A project may contain a single file or many files for translation into one language or several languages. It may also contain reference material, translation memories, term bases, auto-suggest dictionaries and instructions for translators"[1].

Translators may work on a "project package" prepared by the LSP or may create a project themselves, adding their own language resources (translation memories and term bases) or resources provided as separate files by the LSP.

SDL Trados Studio can be extended with plug-ins, developed by SDL or third parties, which provide additional functionalities to perform specific tasks (e.g. file analysis or conversion) or connect to external language resources (e.g. TM servers add MT servers). Typically, these plug-ins can be activated for individual projects thus allowing translators to choose which resources to use for each assignment.

In SDL Trados Studio, all files, regardless of their format, are translated with a single editor that presents the text to be translated broken down into segments. Even though basic formatting (bold, italics, font size) is applied to the text, the translation editor does not always display the correct layout and formatting of the source or target text. In order to check where a given segment appears in the text and how it is presented, translators need to activate the

[1] SDL Trados Studio Project Management Quick Start Guide, 2009-2010, SDL plc.

"File preview" feature – i.e. a dockable pane that provides a visual representation of the file (target text is displayed for the segments already translated).

Several users who are assigned different roles within the project may edit each segment. However, translation and editing are performed in different, usually sequential, stages of the project. In SDL Trados Studio, segments can be flagged with a specific status: translated, reviewed, approved, signed-off, etc. The ability to flag segments based on their status and on the user roles simplifies teamwork and project management.

SDL Trados Studio updates the project statistics in real time and makes them available to the user through a specific module or a dockable pane that can be integrated in the main editor view.

1.3 Working with the MateCat Tool

The main goal of the MateCat Project is to develop a collaborative web-based CAT tool that seamlessly merges translation memory and machine translation technology. The MateCat Tool will be based on distributed client-server architecture. Documents to be translated are uploaded to the MateCat server and assigned to one translator or to a pool of translators. Project managers, translators and reviewers will all be able to access and edit the files at the same time, thus overcoming the limitations of the standard sequential TEP (Translation, Editing, Proofreading) model.

For the design of the MateCat Tool, we assume that a pool of translators shares the same TM. Hence, each segment translated by each translator will become instantly available to all translators working on the same project. As MT will work in tandem with the TM, translators sharing the same TM will also simultaneously share the same MT engine. Similarly to the TM, each update of the shared MT engine will be immediately available to all translators. The MT engine is specific to the project but unaware of the individual translators and documents. Document specific information can, however, be included in the translation requests as detailed below. The presence of several concurrent users is only evidenced as an increase of the required throughput.

Assuming a pool of up to 40 translators, and a 10-word/min individual productivity, the expected data throughput for the MT engine will be around 400 words/min, which should be possible by using a single multi-threaded machine[2].

Now, we will see how the above general requirements relate to the requirements for the self-tuning, user-adaptive, and informative MT functionalities.

1.3.1 Self-tuning MT

Three adaptation modalities are performed at different moments of the workflow.

Domain adaptation is performed at project start time, as soon as the TM and the project documents are available. In this phase, the TM can be used to adapt a generic MT engine, and/or to automatically retrieve/select additional domain-related training data. Domain adaptation is performed only once, offline, and should require not more than a few hours. Ideally, domain adaptation should only require processing of in-domain data. Domain adaptation returns the MT engine models that will be shared by the users. These models will be regularly updated when project adaptation is performed.

Project adaptation is performed during the project, whenever translation of a document is completed or, in general, the users produce a significant amount of parallel data. Project adaptation incrementally refines the existing MT models, by using fresh supervised training data produced by the project. This step should be fast (in the order of minutes) and should only process the newly available data (i.e. with incremental learning methods). Once project adaptation is performed, the shared MT models should be updated so that the effect of adaptation becomes available for all users. As such updates could occur quite often, it should be carefully considered how and when the MT engine should switch to the new models, in order to avoid interference with the work of the pool and with the other adaptation mechanisms.

Document analysis is carried out once the user uploads a document. This analysis generates context information for each sentence. The MT engine should consider this information in order to improve translation coherence. At this time it is not clear how to represent such context information and where to store it. A possibility is that context information is automatically extracted before and during the translation process by an external module, which also attaches this information to the sentence before it is dispatched to the MT engine. As-

[2] Our target architecture on the Amazon Elastic Compute Cloud is a High Memory Double Extra Large instance (13 EC2 compute units, 34Gb RAM, 850Gb HD).

suming that segments of documents are processed in a strictly sequential way, cross-sentence dependencies within a document could be modelled by a directed acyclic graph. This graph should be produced and stored by a specific module that on demand extracts the relevant piece of information to be provided by the MT engine. A natural place to store such information is the CAT tool database, which maintains the progress of work on each document.

1.3.2 User-adaptive MT

This includes the following functionalities that are active while the user is translating.

On-line learning aims to improve the MT on a sentence-by-sentence basis. Implicit and explicit feedback of the user is exploited to adapt the MT models. On-line learning should be performed in almost real-time, so that the effect is visible on the rest of the document. Again, the implementation should manage a stream of feedback arriving to the adaptation module according to a throughput comparable to that of the MT engine. On-line learning should require strictly sequential processing of the feedback stream. From an implementation point of view, the MT engine API will allow the CAT tool to feed the MT engine with feedback from the user, which will be processed by an on-line learner (User Adaptive module).

Context-aware translation takes into account context information that is passed to the MT engine together with the sentence to be translated. The MT Module should be able to process and take advantage of context information.

Informative MT, in first instance performs some post-processing on the MT output in order to compute confidence scores and to point out reliable portions of the MT output. The CAT tool then shows this information to the user.

1.4 Conclusion

In this section we have provided background information about CAT technology and workflow generally applied in the translation industry. By keeping in mind technology standards and common practice, we have then outlined the main user and functional requirements of the MateCat Tool and, in particular, of the novel MT functionalities that we will develop in the project. Finally, the analysis of requirements is complemented by two appendixes, at the end of this report, that provide specifications of the MateCat Tool architecture and a schedule for the development of the first version of the MateCat Tool.

2 First Field Test

During the MateCat project, field tests will be run to evaluate the utility and usability of improved versions of the MateCat Tool. Utility evaluation will be based on key performance indicators (KPI) that will compare productivity of real users employing the MateCat Tool with and without the new MT functionalities developed in the project.

The aim of the first field test was to establish a reference baseline for the performance evaluation of the MateCat Tool. The considered reference baseline is a commercial CAT tool (SDL Trados Studio) integrating a commercial MT engine (Google Translate) and the same translation memory (TM) technology (MyMemory) that will be employed in the MateCat Tool. In the first field test, we tried to automatically measure productivity of the translators in order to estimate the utility of suggestions coming from the MT engine. In addition, this first field test also served the purpose to check the evaluation procedure and to spot potential technical issues.

2.1 Scope and limitations

The first field test was carried out in an uncontrolled environment using standard software adapted in order to fit the need of the field test. This setup presented a number of limitations affecting both the translation process and the data collection and analysis. Also, due to such limitations, the data collected only refers to the drafting phase of a translation.

Translators were required to use a standard version of SDL Trados Studio extended with a plug-in developed by one of the partner of the consortium. It was designed to provide TM matches from the MyMemory server and MT matches from Google Translate. Also, the plug-in allowed collecting data for the field test such as the time spent editing a segment and the match similarity of any matches to the translated segment.

Unfortunately, the plug-in could not record all interactions with a given segment due to limitations imposed by the SDL Trados Studio SDK on which it is based. It only allowed to record actions such as the opening and saving of a segment, the content of the source segment, the best ranking suggestion provided by the plug-in and the target segment saved by the translator. Hence, the plug-in could not tell whether the translator was effectively working on a segment or had stopped working, nor could it detect whether the translator was getting matches from other sources.

In order to overcome such limitations, translators were provided with specific instructions and the collected data was filtered to remove irrelevant data. Sections “Data collection setup”, “Data collection issues” and “Filtering procedure” provide further information on these aspects.

This first field test proved an effective testing ground for the next field tests to be carried out throughout the duration of the MateCat project. Limitations such as those encountered during this field test will be overcome by using software entirely developed by the consortium for the MateCat Tool.

2.2 Defining the KPIs

Two key performance indicators have been identified, that will be used throughout the project:

1. Post-editing effort, which is the average percentage of word changes applied by the translators on the suggestions provided by the CAT tool.
2. Time to edit, which is the average translation drafting speed by the translators.

The first KPI measures the quality of the matches provided by the TM and MT. This corresponds to computing a distance score between matches provided by the system and the post-edited version submitted by the user. The KPI computes a percentage of edit operations performed in the whole set of translated segments.

The second KPI provides information on the quantity of words that are translated in a given time interval as measured in seconds per word. This allows measuring the achieved overall productivity as well as productivity gains.

2.3 Field test data

To set the initial reference baseline, a field test was carried out for the following language pairs and domains:

- English to Italian (EN>IT) – Legal
- English to Italian (EN>IT) - Information Technology
- English to German (EN>DE) – Legal
- English to German (EN>DE) - Information Technology

For each language pair and domain, a team of translators worked on a set of files using SDL Trados Studio and an ad hoc plug-in which connects to the TM server MyMemory. Half of the files were translated with TM matches provided exclusively by MyMemory, without any further suggestions from local translation memories. After completing the first part of the test, translators were sent the second set of files to be translated with a combination of both TM and MT suggestions from the MT engine that was connected to MyMemory.

For the Legal domain, two different documents (.DOC files) were used, one for the TM test and one for the TM+MT test. The two documents contained English text extracted from a call for tender by European institutions that describes the contract binding the tenderer (requirements, selection and exclusion requirements, payments, etc.). As these were standard documents from European institutions, portions of the source text (standard wording) were already available online.

For the information technology domain, RTF files from a software user manual in English were used. The manual was split into two parts of comparable sizes, to compare translation productivity with TM and with TM+MT. The user manual is not publicly available online in English nor in any other language.

2.4 Data collection setup

The field test was carried out using a specific configuration in SDL Trados Studio as described below. Translators were instructed to follow specific rules in order to avoid any potential errors due to improper use of the software measuring performance.

Translators were asked to create a project package in SDL Trados Studio for each test (TM and TM+MT). The project package contained the file(s) to translate and a single translation memory or machine translation provider, that is the MyMemory plug-in. Translators were required not to add any other local TM or MT providers so as to make sure that any matches came exclusively via the MyMemory plug-in - i.e. TM matches from the MyMemory server and MT matches from Google Translate.

Translators were provided only with TM matches for the first part of the test, while for the second part they received matches from TM and MT. The MyMemory server was set up so as to provide TM or TM+MT matches based on the type of test and on the translators' username and IP.

Translators were instructed to translate all segments sequentially: they were asked to not move to a new segment without having completed and saved the current segment. This requirement was meant to avoid issues such as measuring editing time of overlapping segments (i.e. segments enclosing entirely or partially other segments).

2.5 Data collection issues

While translators were interacting with the CAT tool, the following data and statistics were automatically collected for each processed segment:

- Matches provided by the TM server (if any),
- Matches provided by the MT engine (if any),
- Matches used by the translator as a basis for their translation (if any),
- Target segments edited by the translator,
- Time needed to edit each segment.

Unfortunately, the technology used for the test imposed some limitations on the data collection. The SDK for SDL Trados Studio does in fact not allow to develop a plug-in capable of handling all interactions with a given segment.

The plug-in was unable to record when the focus moved from one segment to the other; it could only record opening and saving of a given segment. This limitation lead to an overlapping of the time intervals registered for a number of segments.

Every time a segment is opened in SDL Trados Studio, a GET request is sent to the server in order to retrieve matches from MyMemory. When the translator saves the segment, a SET request is sent and the translated segment is saved back to the server.

Overlapping occurred when translators opened a segment in SDL Trados Studio (GET) and then moved to the next segment without saving the first opened segment (no SET issued). Two specific overlapping cases were identified and removed:

- Enclosure = Segment A GET - Segment B GET - Segment B SET - Segment A SET
- Pipeline = Segment A GET - Segment B GET - Segment A SET - Segment B SET

2.6 Filtering procedure

In order to remove meaningless data, we assumed that translation times shorter or longer than two given thresholds were not related to the translation work flow, but more likely dependent on errors or on the translators' behaviour (e.g. translators who stopped translating for whatever reason without saving the segment they were editing).

The following thresholds were applied to filter reliable segments:

- ≤ 30s per word: Translation times over 30 seconds/word for a drafting of the translation are assumed to be dependent on factors unrelated to the complexity of the source text and more likely dependent on software errors or translators' behaviour (pauses, distractions, etc.).
- ≥ 0.5s per word: Translation times below 0.5 seconds/word are assumed to be unrealistic for most segments and result of an accidental interaction with the software (e.g. saving a segment without reading or editing it).

Collected data was also filtered to remove all 100% matches and repetitions: the time to edit for those segments is irrelevant since SDL Trados Studio automatically translates segments identical to matches provided by the TM without any human interactions.

Three protocol violations were identified which resulted in the removal of a number of segments:

1. One translator (EN>DE Legal) used an improper set up in SDL Trados Studio resulting in a loss of most segments[3]. All data from this translator were removed from the data set.
2. Two translators (EN>IT Information Technology and EN>IT Legal) received MT matches while working on the TM-only test. This violation was caused by an improper configuration of the filter on the MyMemory server. All data from both translators were removed from the data set.
3. Translators did not always translate sequentially (overlapping - see "Data collection setup" above). All data from one translator (EN>IT Information Technology) and a number of segments from all translators were removed from the data set.

[3] Translated segments were not sent to the MyMemory server.

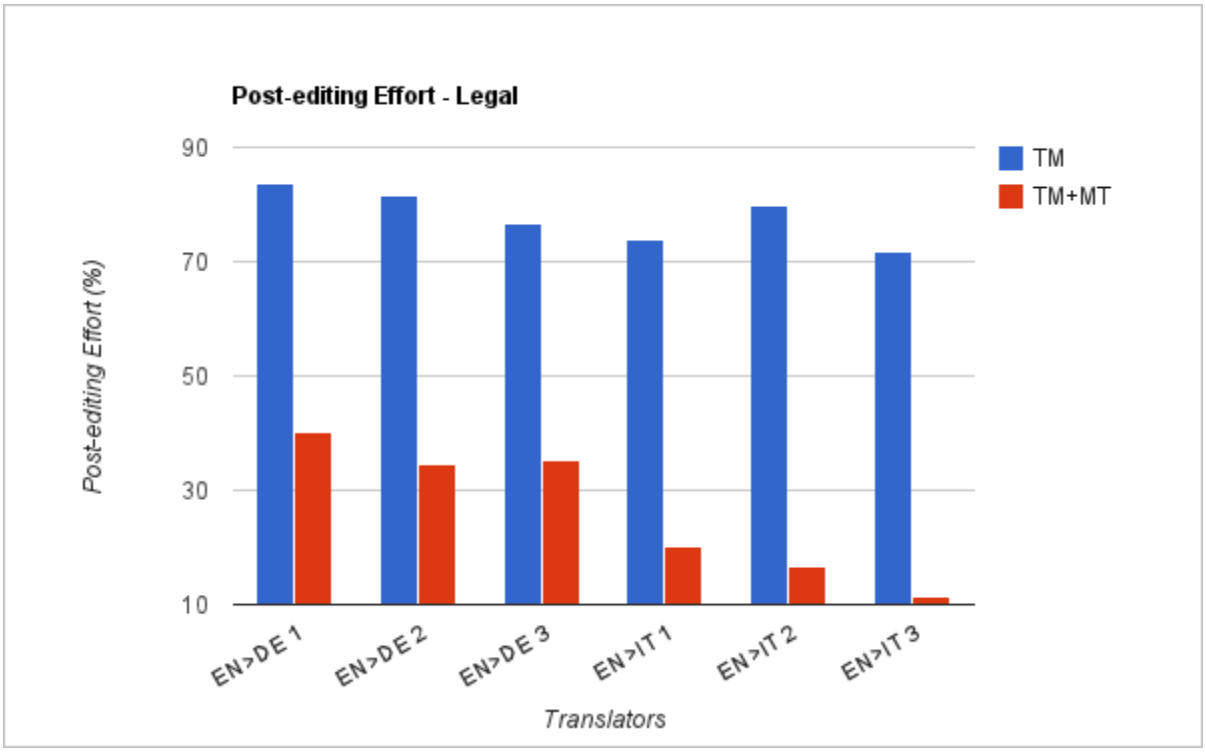
After applying the thresholds and deductions the number of words available for the statistical analysis is as follows:

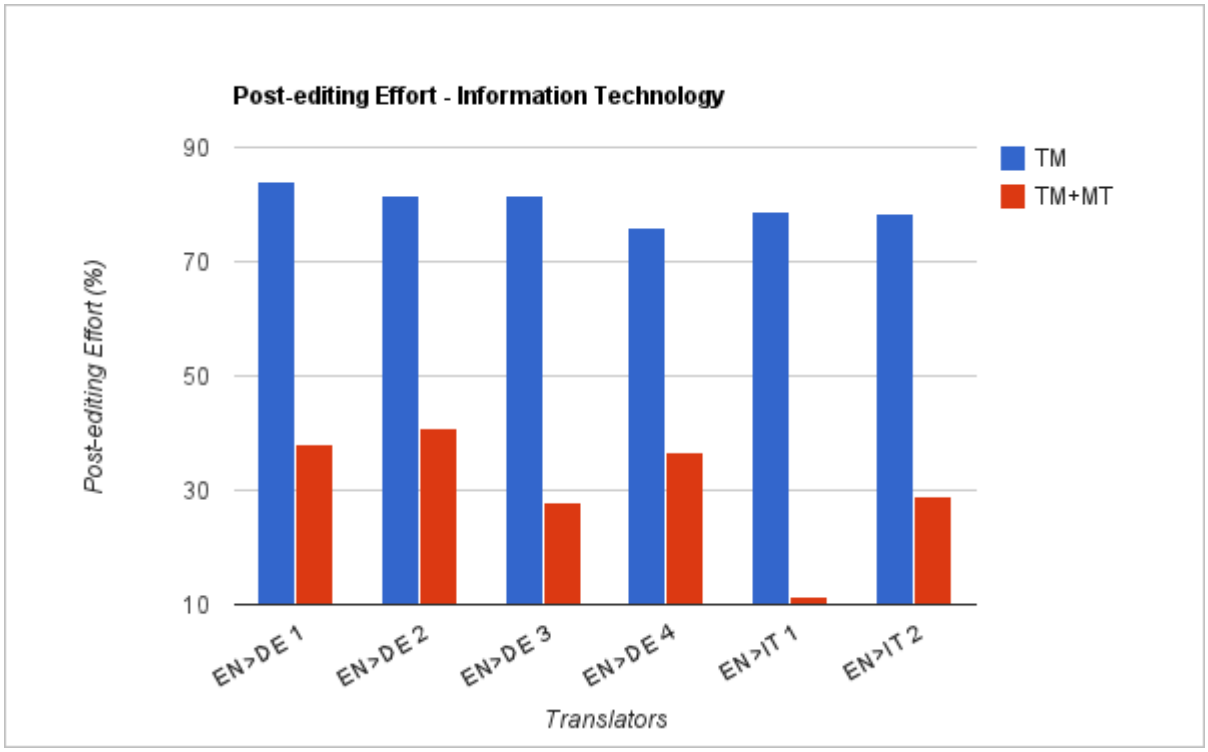
	EN>DE	**EN>IT**	**Total**
Legal (TM)	7,221	7,041	**14,262**
Legal (TM+MT)	8,568	13,087	**21,655**
IT (TM)	18,425	8,553	**26,978**
IT (TM+MT)	19,972	9,791	**29,763**
Total	**54,186**	**28,472**	**92,658**

2.7 Results on KPI 1: Post-editing Effort

The first KPI aims at defining the quality of the matches provided by the TM and MT systems. We measured the percentage of words edited in a segment by comparing the match provided by the system and the edited segment submitted by the translator. A proprietary function was used which compares two segments and assigns a match percentage based on factors such as same words in the two segments and word order. Applying penalties based on factors such as formatting, tags, casing, etc then alters the match percentage. The function is designed to be as similar as possible to the matching algorithm used in commercial CAT tools. The similarity match can be interpreted as an indication of the quality of the suggestions provided by the TM and MT systems. On the other hand, an estimate of the involved post-editing effort can be simply computed by taking the complement of the similarity match (100%-SimilarityMatch).

In the performed field test, the overall post-editing effort decreases significantly when adding MT matches to the matches provided by the TM. Even though this may be considered an obvious consequence of using two sources for the matches, the extent to which the quality improves proves the effectiveness of the MT engine used in the test (Google Translate). The following figures show the average percentage of post-editing effort resulting from using the TM alone and a combination of TM and MT, for each language pair and domain.





Post-editing effort for matches from TM and TM+MT.

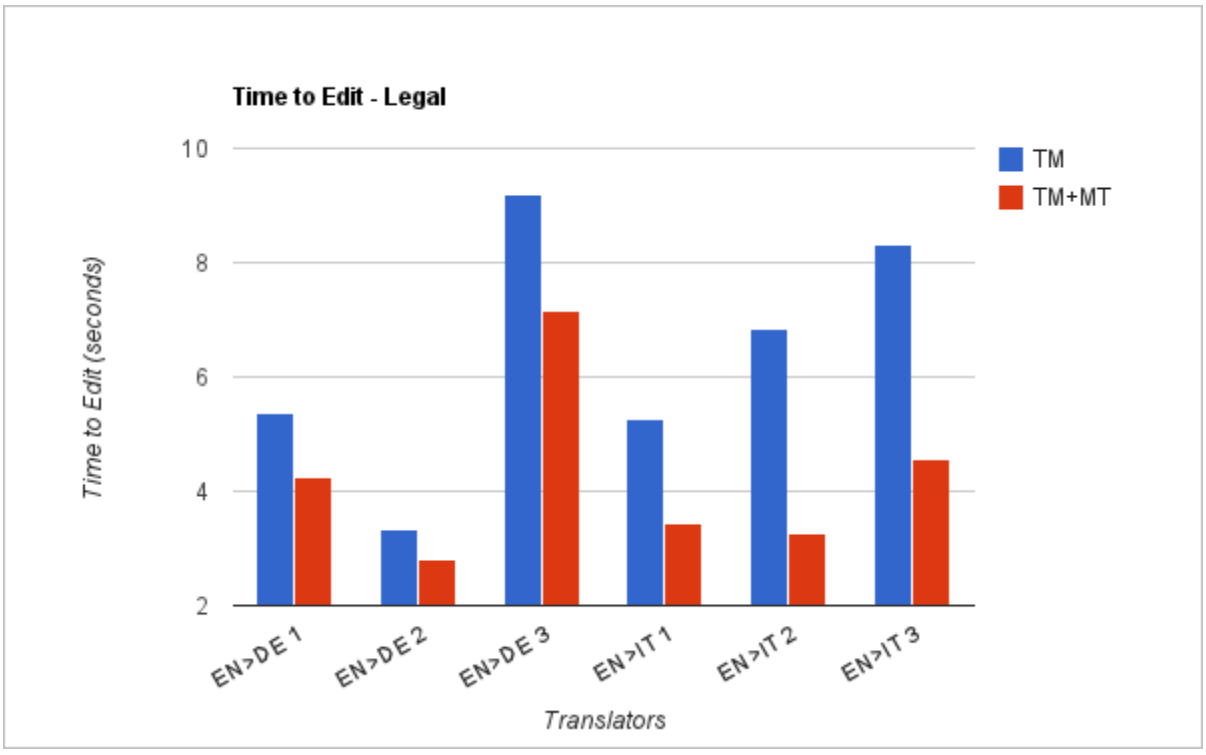
From the above charts, it results that individual translators took advantage from suggestions coming from the MT engine. On the legal domain, post-editing effort with only TM was on average 80.7% for EN>DE and 75% for EN>IT. With the availability of MT suggestions, these figures dropped on average, respectively, to 36.7% and 16.15%. The relative gain in

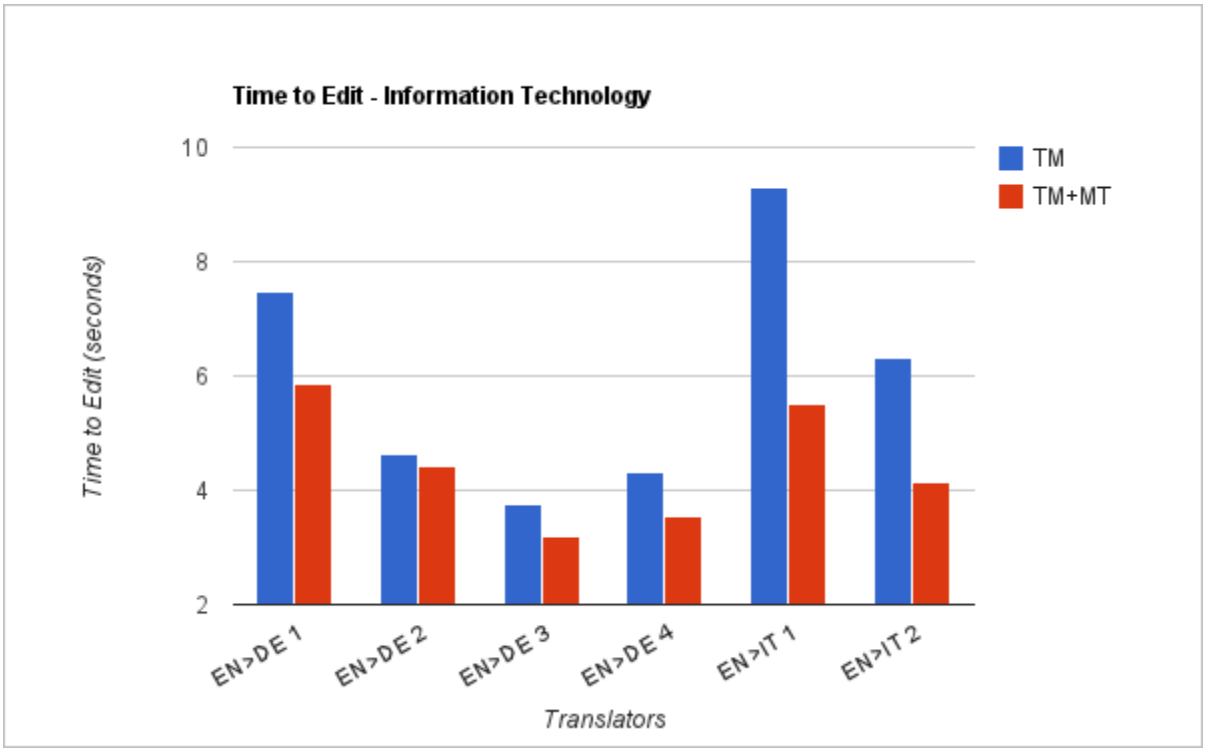
post-editing effort on the legal domain results hence of 54.6% for EN>DE and 78.5% for EN>IT.

On the information technology documents, post-editing effort with only TM was on average 80.9% for EN>DE and 78.6% for EN>IT. With the availability of MT suggestions, the corresponding figures dropped to 35.9% and 20.2%, respectively. Hence, the relative gains on the two translation directions were 55.5% and 74.2%.

2.8 Results on KPI 2: Time to Edit

The second KPI aims at measuring the average productivity of translators getting only TM matches as compared to translators getting also suggestions from MT. In particular, we measure the average time taken by the translator to complete a segment in seconds per word. We expect this indicator to be related to the one described in the previous section. In other words, improvement in the quality of the provided matches should directly affect the performance of the translators. The following charts show that on both domains and language pairs most of the translators were able to achieve substantial improvements in productivity.





Average time to edit per word of translated segment.

From the above charts it emerges that all translators reduced their time-to-edit figures when passing from the TM to the TM+MT suggestion mode. In general, time-to-edit figures varied significantly across translators, languages, and domains (some possible explanations are provided in the following Discussion section). In this respect, it seems more appropriate to focus on the relative time-to-edit gains achieved by each translator and to compute averages over such figures. For the legal domain, the average relative time-to-edit gains are 19.5%, for EN>DE, and 44.0%, for EN>IT. For the information technology documents, the average relative gain are 14.4% on EN>DE and 37.7% on EN>IT.

2.9 Discussion

Even though all translators translated the same content and were provided with the same instructions and information, the results for the two KPIs show a certain degree of variation in terms of time to edit and post-editing effort. The variation depends on two factors: the quality of the matches provided by the plug-in and the performance of each translator.

The quality of the matches from the MyMemory TM server depends on the amount of translated segments that it contains for each language pair and domain. There are some differences in the number of segments for EN>DE and EN>IT. Also, MT matches tend to be of higher quality for the EN>IT language pair than for EN>DE.

All translators were supposed to deliver a drafting of the target text. However, it is generally difficult to assess objectively the quality of a translation and translators are not capable of determining when their translations are "good enough" for a drafting. Some translators may consider it appropriate to deliver a translation that is semantically correct while poor in style. Some others may put in more effort in order to provide not only a semantically correct translation, but also one they consider more appropriate from a linguistic (i.e. grammar, style) and a terminological point of view.

The different approach by each translator played a role in the variations we can see in terms of time to edit and post-editing effort. Some translators accepted MT matches without much editing because they considered such matches to be semantically correct and appropriate for a drafting of the text. Others spent more time on each segment editing more words because they felt they needed to provide a higher quality target text (improving on style and language quality).

Moreover, time to edit can also be influenced by how the translators use the software (SDL Trados Studio). While all translators were required to use the same settings for the project package, we couldn't force them to use a specific setting for the UI (in SDL Trados Studio the UI elements can be re-arranged to match the translator's requirements). How UI elements are arranged can affect performance: translators may have to perform some extra actions in order to view the matches from the TM or MT (if the translation matches window is too small, translators are required to scroll through the results using a mouse or touchpad), they may have to activate the preview feature to see the text they are translating in context (although this may not be too important when working on a draft). Also, some translators may be used to move from one segment to another using keyboard shortcuts, while others may use the mouse or touchpad. Even though such activities do not account for significant changes in terms of overall productivity on a daily basis, they can affect the time to edit by 0.5 seconds per segment.

2.10 Conclusion

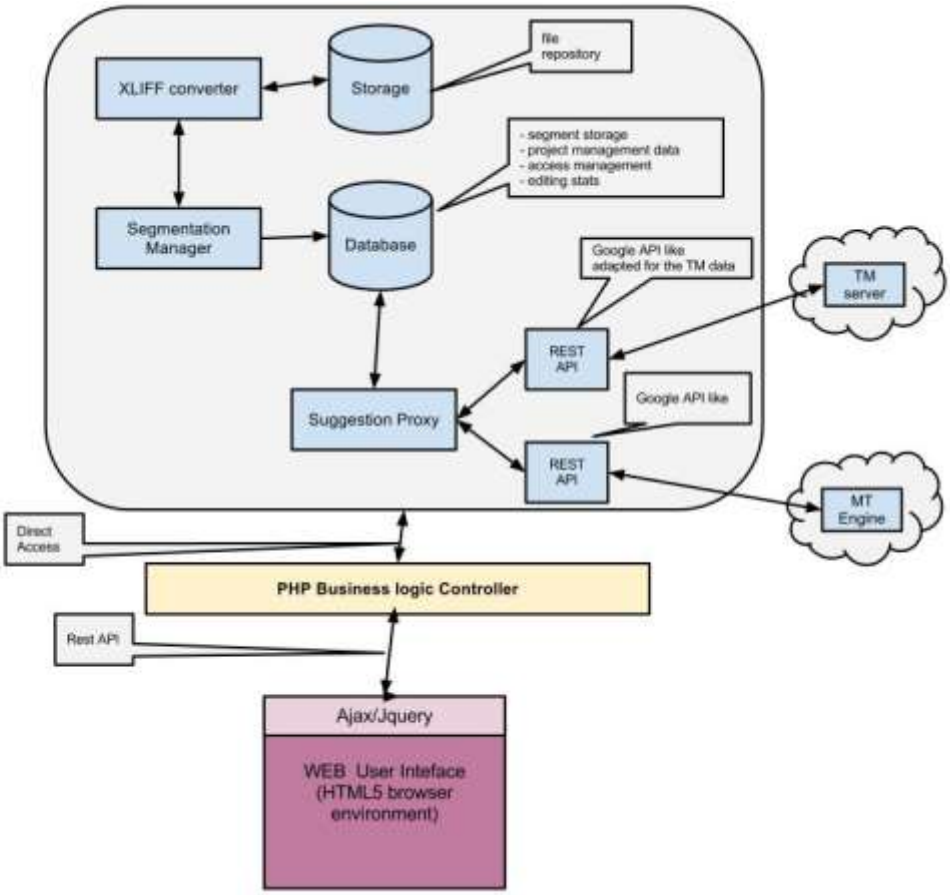
The first field test provided useful insights about the KPIs we decided to adopt in MateCat, the protocol to follow when running the test with the translators, and the way to collect and process the data from the users during the next field tests. These insights will be explicated in the forthcoming Deliverable 5.2. Evaluation Plan, which will describe the evaluation methods and protocols followed to measure progress in the project.

3 Appendix 1. Architecture Specifications

The following diagrams and notes report recent progress in defining the overall architecture and the single modules/components of the MateCat Tool. More detailed descriptions and specifications are circulating in the consortium as internal documents that will be added to the final documentation of the MateCat Tool, once its software will be publicly released.

3.1 MateCat Tool Architecture

The diagram below integrates the description of the architecture of the MateCat Tool given in the Description of Work (DoW).



The MateCat Tool will be implemented with a distributed architecture, which will allow multiple users to use the tool concurrently through a web user interface that connects to the CAT server via a PHP controller.

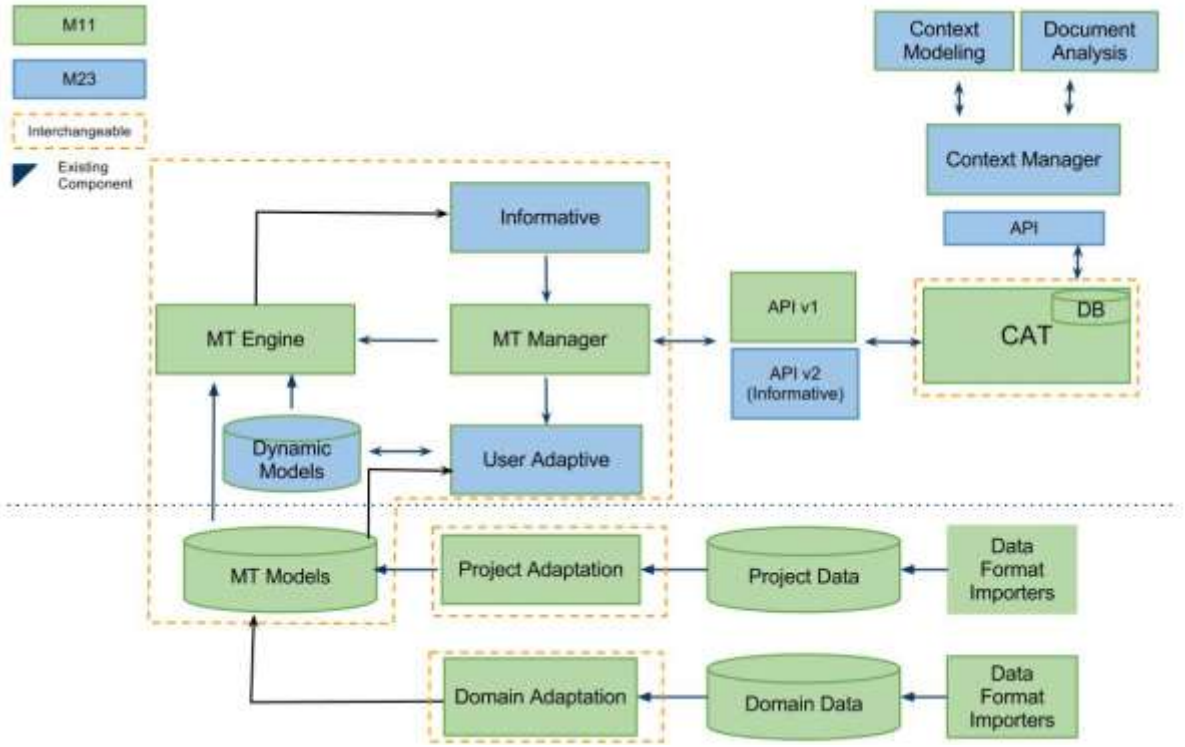
The XLIFF converter module will be used to convert all translatable files into an XLIFF file[4]. The content of the file will be segmented by the segmentation manager and stored in the database, which will contain the segments extracted from the files for each project. The segments will hold information on the language pairs, on the users that will take care of the translation/revision, on the status (draft/translated/revised) and on the editing stats. Each segment from the database is sent to the suggestion proxy and from there to the TM and MT servers. Suggestions from the TM and MT are sent back to the suggestion proxy, which ranks them based on a dynamic matching algorithm.

The MateCat Tool will connect to a TM server and an MT engine via two APIs based on and extending the Google Translate API v2[5]. The first API will manage the communication with the MT engine providing two functions: GET (i.e. request of a translation to the MT engine) and SET (i.e. feedback to the MT engine for the sake of adaptation). The second API will manage the communication with the TM server. In this case, more information is needed to identify the specific translation memory (if provided) and to perform the two operations GET (search entry in the TM) and SET (add entry to the TM).

[4] XLIFF (XML Localisation Interchange File Format), an XML file for exchanging localization data.
[5] https://developers.google.com/translate/v2/getting_started

3.2 Context Manager Module

In order to generate and update context information at the document level, a Context Manager (CM) will be integrated into the CAT server. The CM Module will interact with the CAT Tool database that will store and keep track of all documents and segments processed by the users working with the MateCat Tool.



The CM Module is responsible for:

- Carrying out a preliminary document analysis to detect intra- and inter-segment dependencies;
- Updating context information of segments, as soon as such information becomes available;
- Attaching proper context information to the segments before they are sent to the MT module.

Type and representation of dependencies as well as means to integrate them in the MT decoder will be investigated within Task 1.3 (document analysis) and Task 2.2 (context -aware translation).

3.3 MT Module

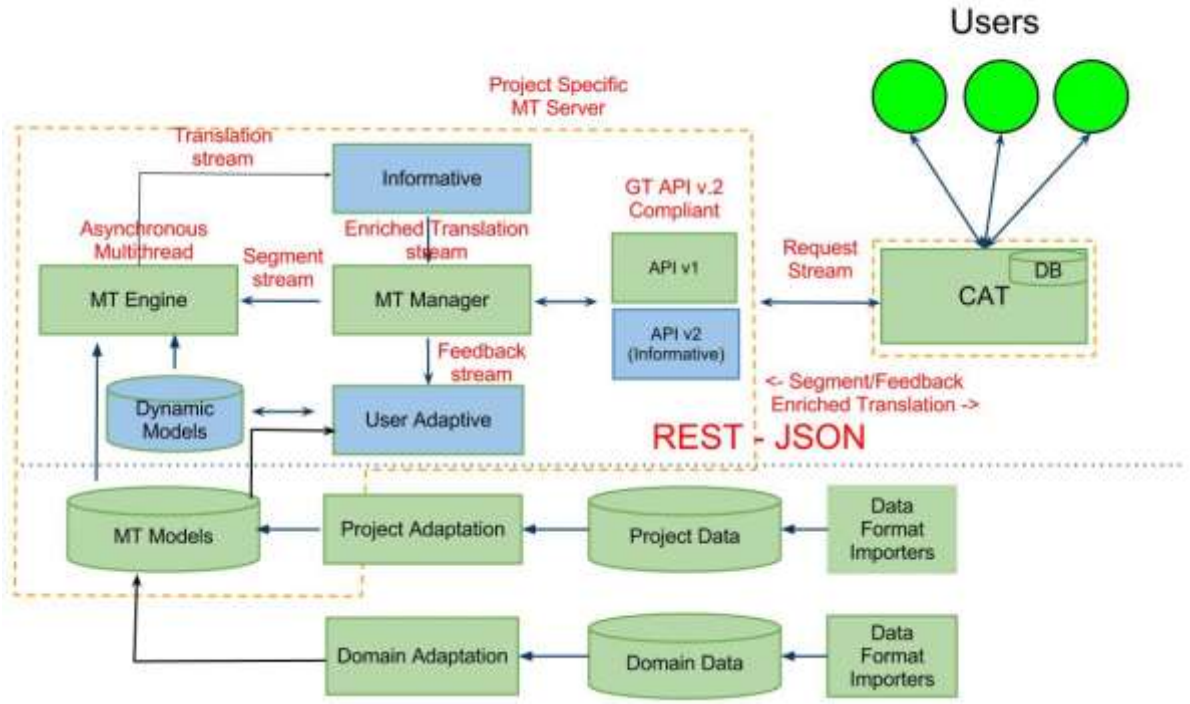
The proposal presented in the Description of Work has been updated to permit sharing of the same MT engine among different translators working on the same project or in general with the same TM. In the new architecture, the MT engine serves a stream of segment translation requests coming from several users and it does not maintain any information about the documents nor the users. Finally, the Context Manager has been moved outside the MT Module.

Besides segments to be translated, the MT Module receives and processes other types of information:

- Context information attached to segments, that is directly exploited by the decoder;
- Segments provided with their translations and other feedback information, that are exploited for on-line adaptation by the User Adaptive component;
- Requests of translation options for single terms inside a segment, which are processed by a specific Informative component.

The services of the MT Module are provided as a Web Service, which will be compliant with the Google Translation API v.2.

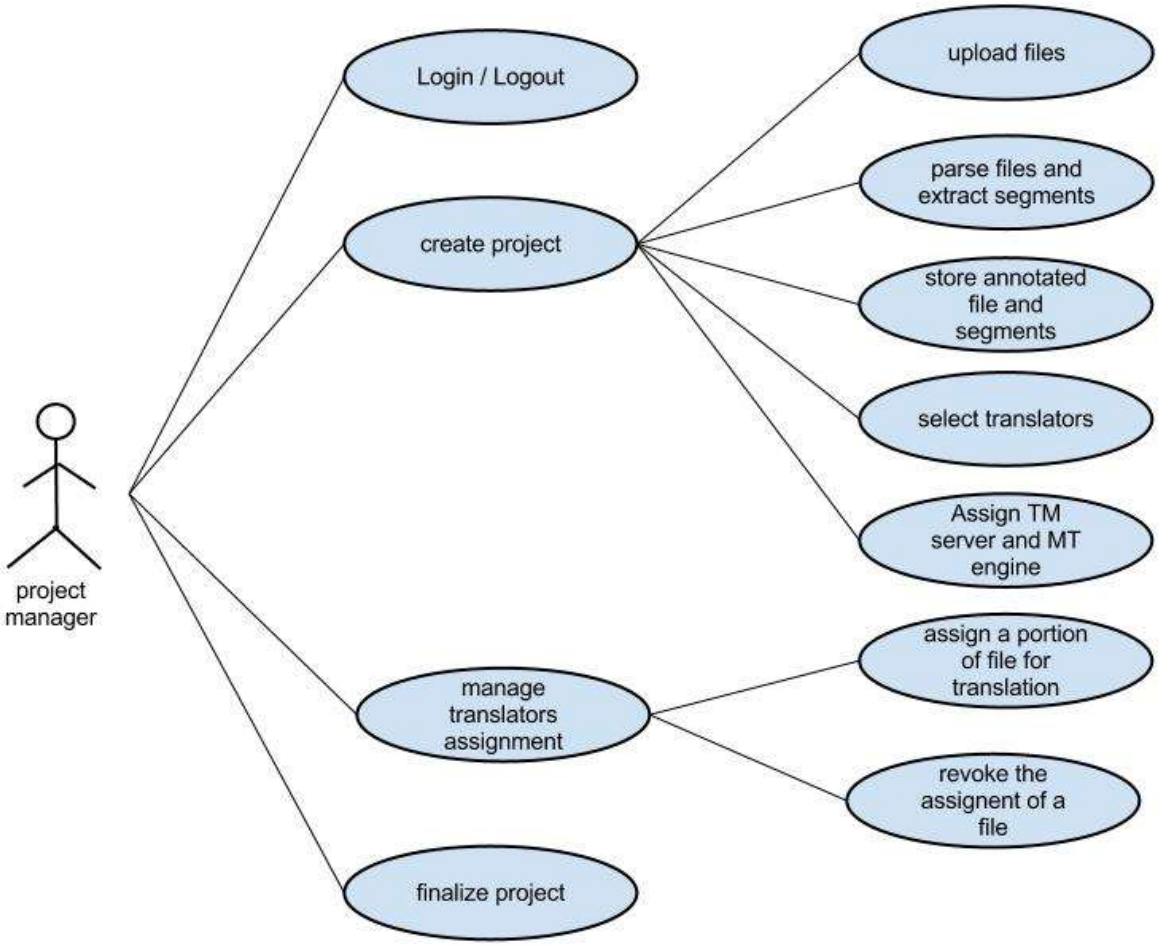
The following diagram shows the data-flow between the CAT Tool and the MT component. The MT engine will process a stream of segments in multi threading and asynchronously, so that translations of individual segments are returned as soon as they become available.



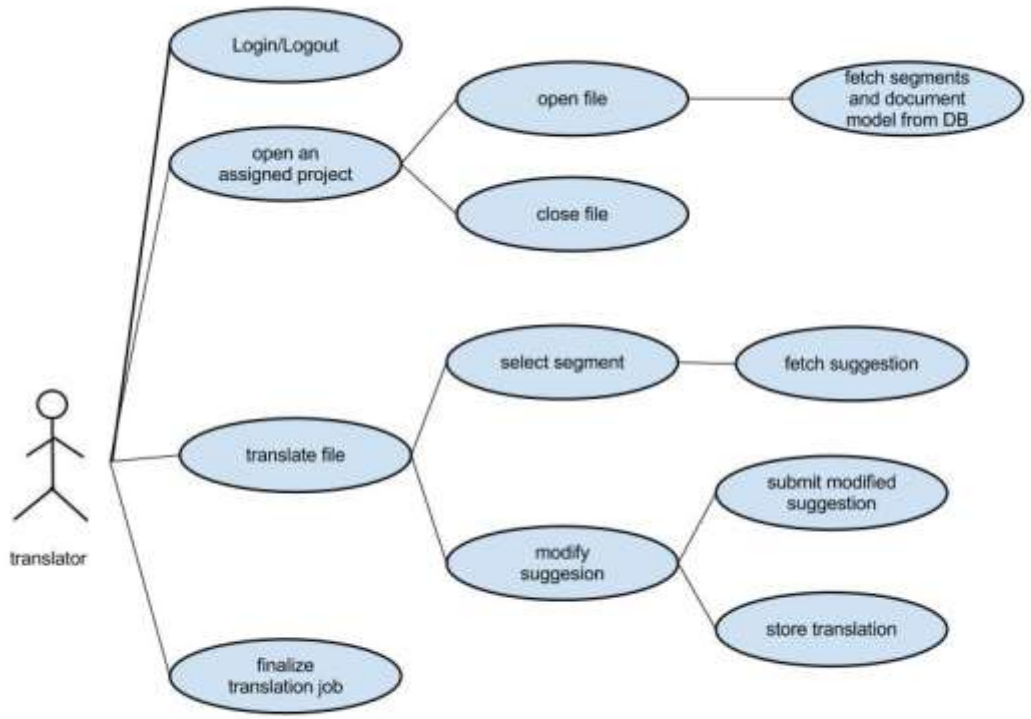
3.4 Use Cases

The following diagrams show the operations that two classes of users (project manager and translator) will be able to perform with the MateCat Tool.

3.4.1 Project Manager



3.4.2 Translator



4 Appendix 2. Development Road Map

The development of the MateCat tool will follow the road map described in the Description of Work. In the following, we provide a more detailed schedule for the development of the first version.

4.1 MateCat Tool v.1

4.1.1 Features

- Basic User Interface. Tool imports simple file formats and mark-up (RTF and XML) and converts them internally into the XLIFF[6] format.
- Data stored in the CAT tool and in the TM will be untokenized and segmented according to the SRX segmentation[7] format.

[6] XLIFF, the XML Localisation Interchange File Format, is a text format widely used by the localisation industry.

Baseline Definition

- MT for EN>IT and EN>DE on legal and IT domains.
- Domain and project adaptation at reasonable computational costs: 10h and 1h, respectively, on a 8-core 32GB machine.
- MT module: language/project-specific server, Google Translate v.2 API, multi-thread processing, RTF and XML mark-up of segments

4.1.2 Schedule

15 May 2012 Alpha version of MT module	• Baseline translation model for IT domain EN>IT • Basic domain/project adaptation • Google Translate v.2 API compatible • Multithreaded processing of segments
30 June 2012 Alpha version of CAT Server	• Connects to Google Translate and Alpha version of MT server.
31 August 2012 Alpha version of MateCat Tool v.1.	• MT component will manage mark-up • Translation models for both legal and information technology domain • EN>IT and EN>DE • CAT component will feature basic UI • XLIFF importer.
30 September 2012 Beta version of MateCat Tool v.1.	• MT will feature domain/project adaptation from segmented, untokenized data with SRX segmentation • CAT server will feature improved basic UI, XLIFF importer.

[7] SRX, the Segmentation Rules eXchange, is an XML-based standard for text segmentation adopted by the translation industry in order to support interoperability of translation memories.

5 Appendix 3. Data collection and post-editing guidelines

5.1 Issues in data collection

The software set-up used for the baseline definition test put a number of limitations on the data collection task. The MyMemory plugin used for the test was developed using the SDL Trados Studio SDK which only allows a limited integration with the CAT tool and with the MyMemory server. Such limitations prevented us from catching every user interactions with the translation tool: we could not identify when the translator stopped working (pauses) nor log successive edits of a translated segment (this means that we could only log the first draft translation).

5.2 post-editing guidelines

In the following sections, we present the guidelines and instructions provided to post-editors working on the baseline definition project.

5.2.1 General information

Two teams of translators worked on the baseline definition translation project. One team translated legal documents while the other worked on information technology files. The instructions below applied to both teams even though some images refer explicitly to the legal project.

Translators were asked to deliver a draft translation (light post-editing) of the documents assigned to them. This meant a change in their standard workflow in that they were required to translate each segment sequentially and they were not allowed to go back to a previously translated segment. The result is that for each segment they were only providing their first draft without any further editing or improvements.

Translators were instructed to translate two separate jobs (phase 1 and 2 of the test) using SDL Trados Studio 2009 SP3 or SDL Trados Studio 2011 with the relevant MyMemory plugin:

- For the first job, they only received translation memory matches from MyMemory

- For the second job, they received translation memory matches and machine translation suggestions from MyMemory

The Project Manager sent them two separate jobs with two different Purchase Orders. First, they were required to deliver the first job (TM matches only). After the delivery of the first job, the PM sent them the second job (TM & MT matches).

For each job, translators were require to deliver the following files:

- Bilingual files (SDLXLIFF)
- Target files

5.2.2 Note for the Legal translation jobs

Translators received two files:

1 Source file to translate (approximately 6,000 words);

A reference file which contains the text they translated plus the rest of the original document to be used for contextual information.

5.3 Step by step guide

Before translators started working, they were required to prepare the project following the instructions detailed in the following pages.

Here's a step by step guide to the translation process:

1 Installing the plugin (see section *Installing the plugin*).

2 Generating a private key for MyMemory (each translators had to use the same personal key for both jobs). See section *Key generation*.

3 Sending to the Project Manager the username and key that they were using with the plugin. See point 6 in *Project creation in Trados Studio* below.

4 Waiting for the Project Manager to confirm that they could start the translation.

5 Creating a translation project for the first job following the instructions in *Project creation in Trados Studio*.

6 Completing the translation of the first job and delivering it to the Project Manager.

7 The PM would then send the second job.

8 Creating a new translation project for the second job following the instructions in *Project creation in Trados Studio* and using the same username and key that each of them used for the first job (one key per translator).

Completing the translation of the second job and delivering it to the Project Manager.

5.3.1 Installing the plugin

Depending on the Trados Studio version used, translators were asked to install one of the following plugins provided to them via an FTP server:

- *mymemory_plugin_studio2009.exe* (for Trados 2009 SP3)
- *mymemory_plugin_studio2011.exe* (for Trados 2011)

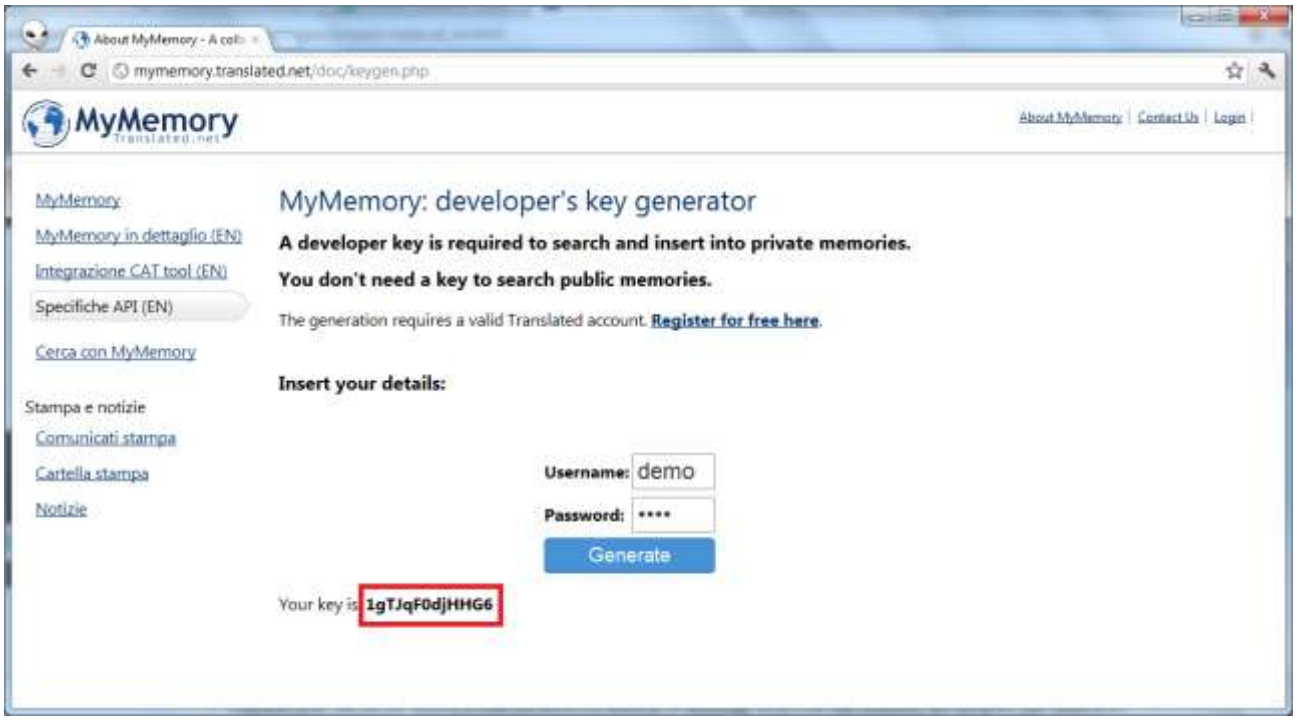
This version of the MyMemory plugin (2.2) is an updated version that has not been released to the public. They were required to install this version even if they were already using the previous version of the MyMemory plugin (2.1).

5.3.2 Key generation

In order to use the MyMemory plugin, translators needed a specific username and key.

Translators could generate their personal key entering their Translated username and password (i.e. the credentials they use to access their profile on Translated.net) and then clicking on the *Generate* button on the following page: http://mymemory.translated.net/doc/keygen.php

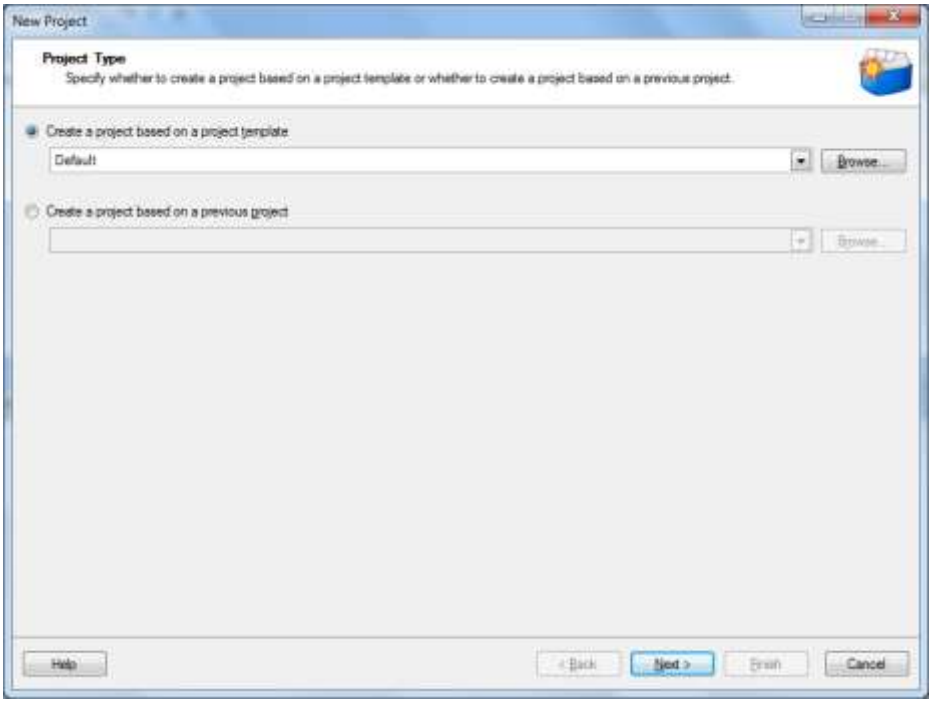
The key would be displayed as shown in the figure below.



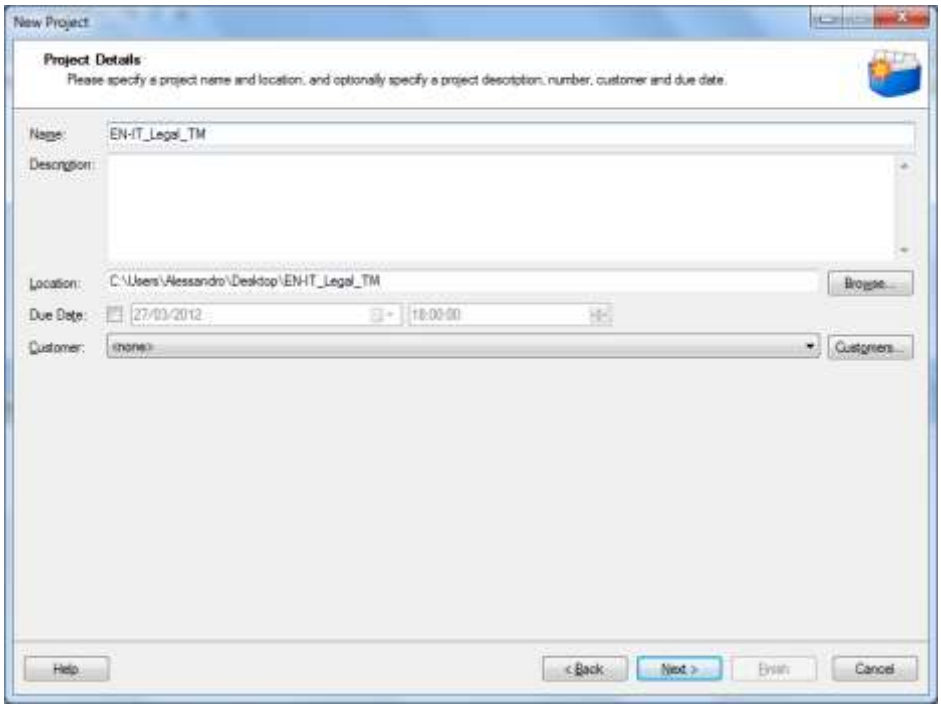
5.3.3 Project creation in Trados Studio

Translators were required to follow accurately the instructions provided below in order to create a valid project in Trados Studio for both translation jobs.

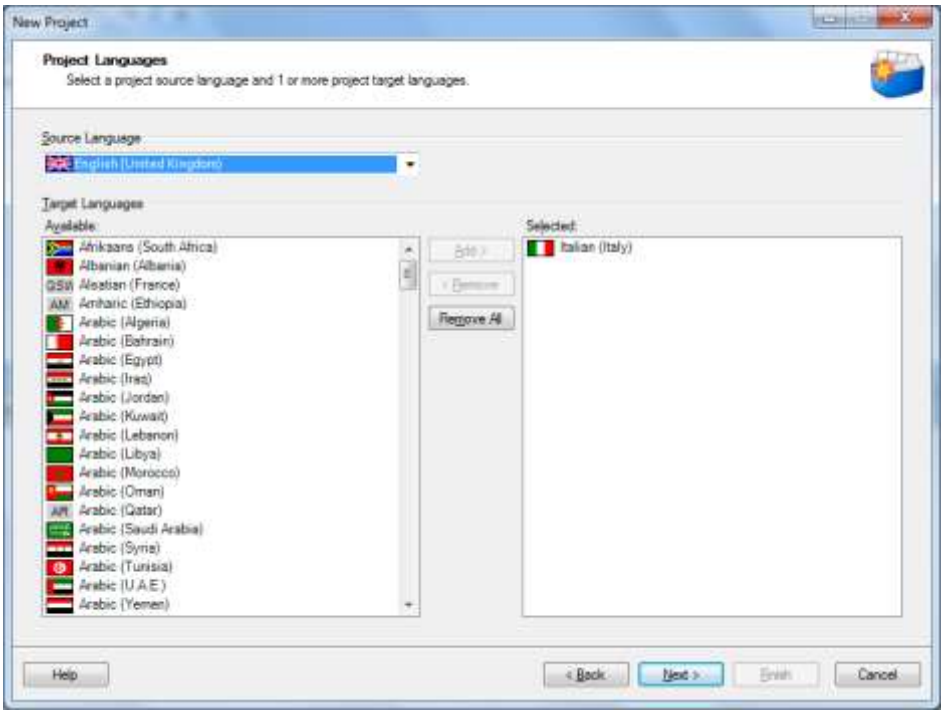
1. Click on *New project* and create a new project using the *Default* template.


New Project
Project Type
Specify whether to create a project based on a project template or whether to create a project based on a previous project.
Create a project based on a project template
Default
Browse...
Create a project based on a previous project
Help
< Back
Next >
Finish
Cancel

2. When assigning a name to the project, use the same name of the folder that the Project Manager sent. For example, for the folder “EN-IT_Legal_TM”, enter the details displayed in the figure below.

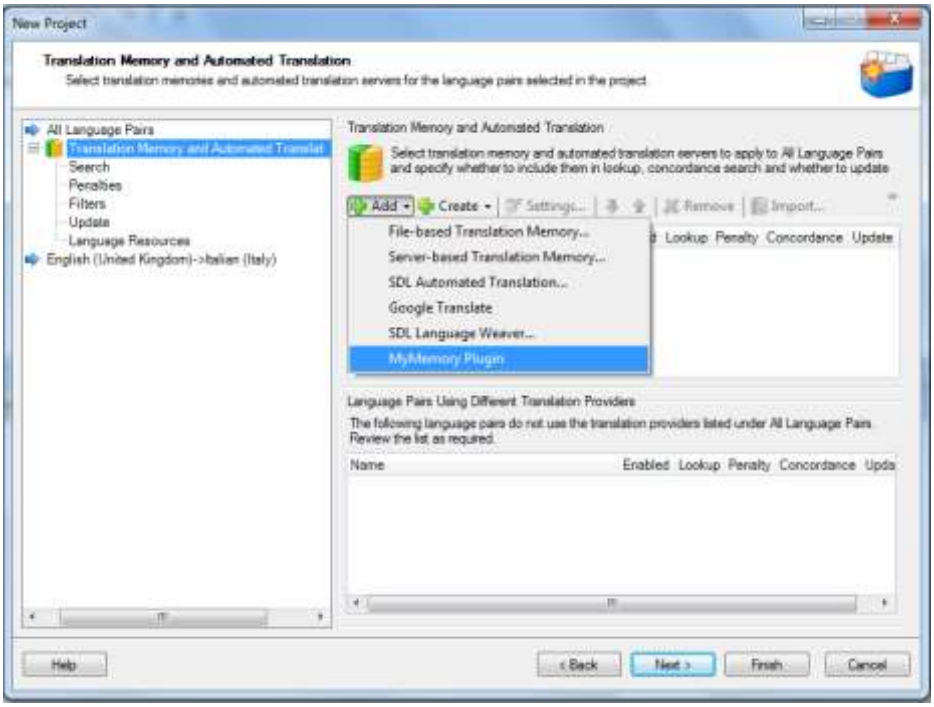


3. Specify English UK as the source language.

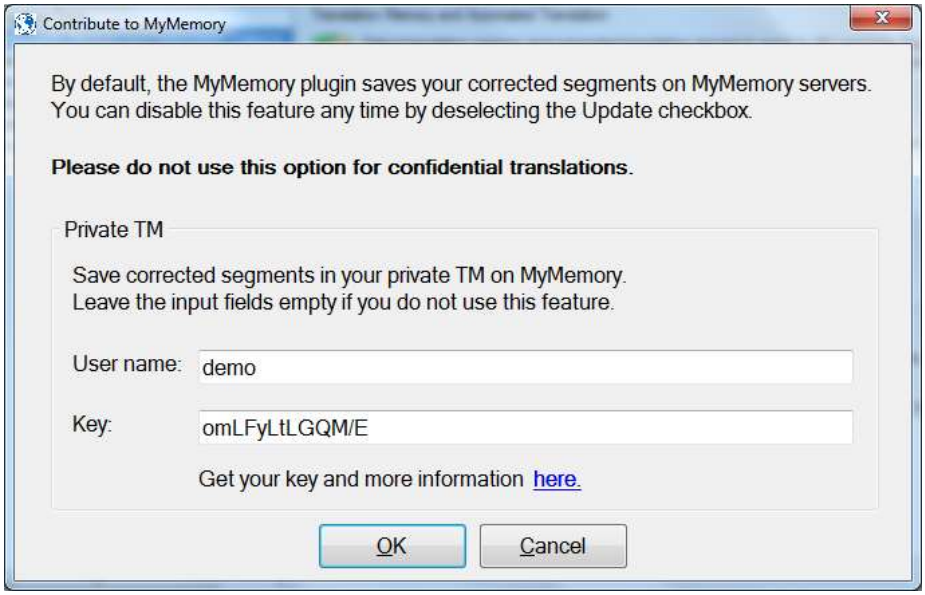


4. Add all the files that need to be translated. Note: for the information technology jobs, all the files were to be added at once.

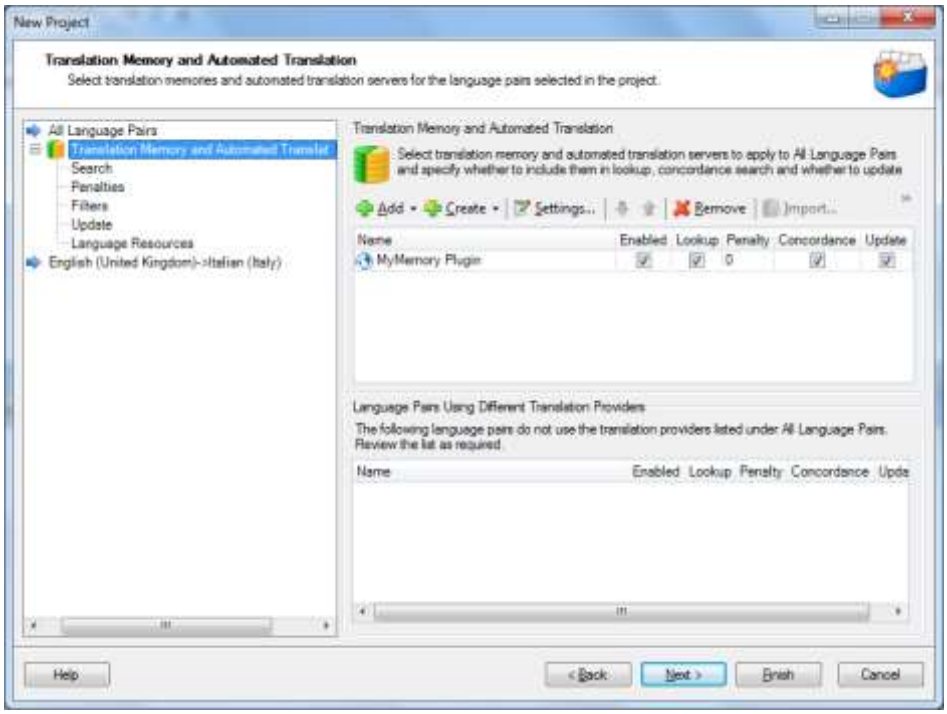
5. The only translation memory to be added had to be the MyMemory Plugin. See section *Installing the plugin* for instructions on how to install the plugin.



6. When adding the plugin, the dialog below would be displayed and translators were required to enter their personal username and key.


Contribute to MyMemory
By default, the MyMemory plugin saves your corrected segments on MyMemory servers.
You can disable this feature any time by deselecting the Update checkbox.
Please do not use this option for confidential translations.
Private TM
Save corrected segments in your private TM on MyMemory.
Leave the input fields empty if you do not use this feature.
User name:
demo
Key:
omLFyLtLGQM/E
Get your key and more information here.
OK
Cancel

7. Make sure that the *Update* feature is active as shown in the following figure.



10. Finish creating the project and start translating.